ECE 445

SENIOR DESIGN LABORATORY

PROPOSAL

Grasping Any Object with Robotic Arms with Language Instructions

---

Team #10

JUNZHOU FANG
(junzhou5@illinois.edu)
JUNSHENG HUANG
(jh103@illinois.edu)
ZIXIN ZHU (zixinz6@illinois.edu)
ZIXUAN ZHANG
(zixuan21@illinois.edu)

TA: Tianci Cai, Tielong Tang

March 14, 2025

Contents

1 Introduction

1.1 Problem

The ability of robotic arms to grasp objects based on human instructions is increasingly vital as human-robot collaboration becomes a popular research and application field. However, this task presents substantial challenges. It requires the seamless integration of advanced computer vision, natural language processing, and precise robotic control to perform accurate grasping actions according to oral instruction. Many existing robotic systems are limited to specific contexts or require extensive retraining for new objects, lacking the generalization needed for a broad object vocabulary. Overcoming these hurdles is essential to creating adaptable robotic systems that enhance human productivity and independence in dynamic, human-centric settings. This task is crucial for the integration of robots in everyday environments, as millions of households and industries are expected to adopt robotic assistance by the coming decades, necessitating intuitive and flexible interaction capabilities.

1.2 Solution

Our expected solution is a smart robotic arm equipped with a well-designed recognition system based on computer vision and natural language processing. The image analysis module of the robotic arm will be trained using RGB images and corresponding captions, allowing it to categorize objects in the robotic arm camera. We will also use a language processing module to extract the name of the intended objects from the input text or voice command. Then, the robotic arm will move along the optimal path to grab the object to the designated position. The visual illustration of our robotic arm is shown below.

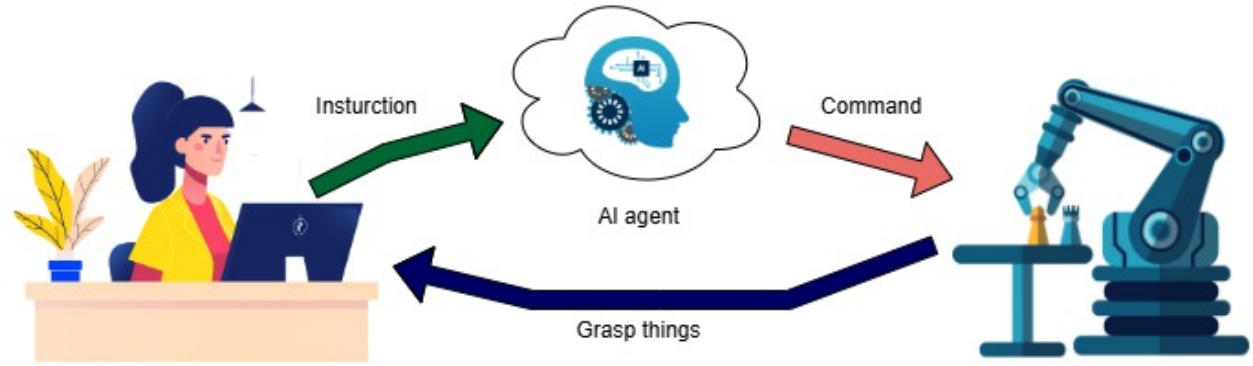

Figure 1: The Visual Illustration

1.3 High-level Requirements List

- Reliability: The system should maintain a high level of reliability, such as the accuracy of 80% recognition when matching input instructions and figures.

- Generalization: The system should support grasping for at least 10 distinct objects and be able to deal with out-of-vocabulary phenomena.
- Efficiency: The system should avoid collisions during the path execution and complete each grasping task in 2 minutes.

2 Design

2.1 Block Diagram

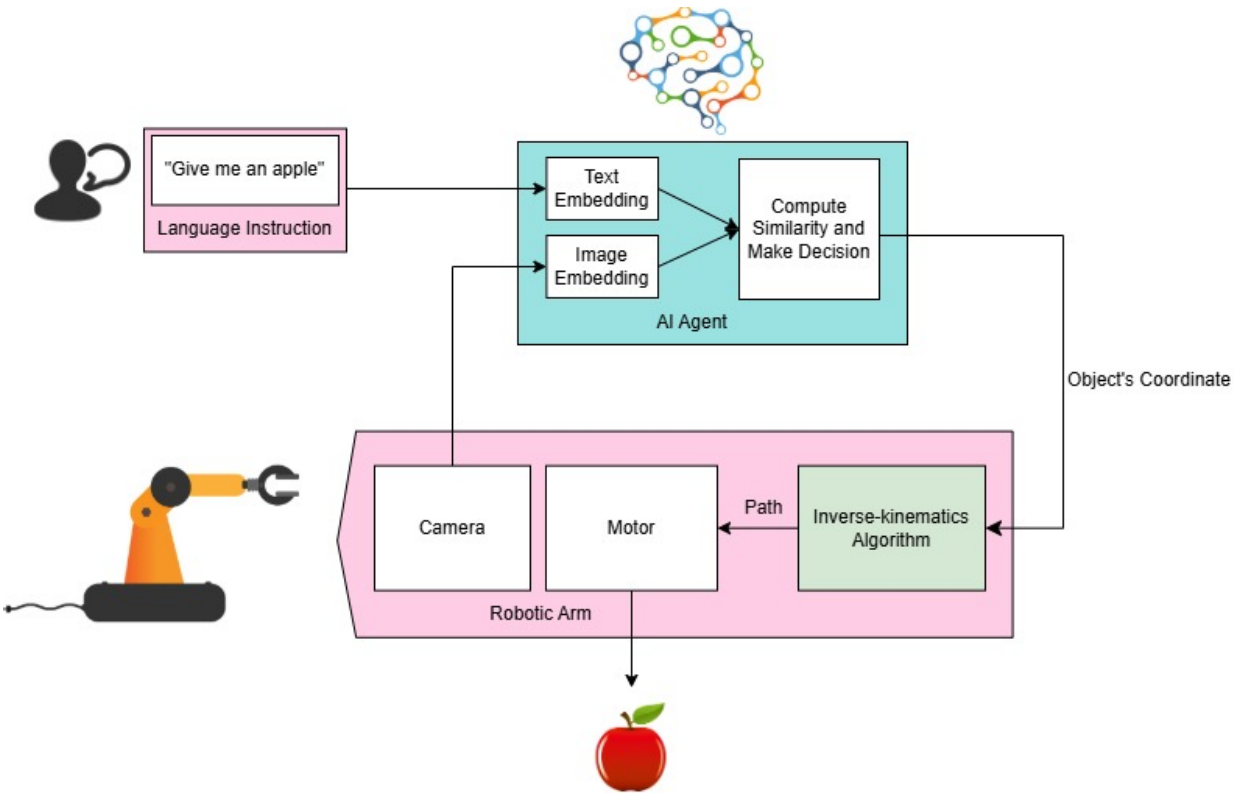

Figure 2: Model Layout

The high-level diagram of our proposal. The AI agent is responsible for object selection, and the robotic arm will fetch the corresponding object according to agent's decision. With more details, after a language instruction sent in, it will be transformed into high-dimensional vectors in the Text Embedding module. At the same time, the image embedding module reads the image taken by the camera on the robotic arm, picks out all the objects, and performs another embedding to get a series of arrays. The agent then compare the similarity between these object arrays and the text vector, and choose the object with the highest similarity score. The position of this object can be obtained directly through the camera, and an inverse-kinematic algorithm is used to compute the movement of each node on the robotic arm. And finally, the arm will grasp the intended object.

2.2 Subsystem 1: AI Agent

2.2.1 Overview

Our AI agent integrates natural language understanding with visual perception to enable precise object manipulation. The agent accepts either text input or speech commands,

intelligently identifying objects for robotic arm grasping task.

The first task is **Automatic Speech Recognition (ASR)**. There are some traditional ways to do this task like HMM-GMM. However, as the deep learning method growing up, end-to-end models seems to have higher performance. Among which, we choose a lightening framework from Meta AI: Wav2Vec 2.0 [1]. It is a self-supervised speech recognition framework that learns powerful speech representations directly from raw audio using Transformer architectures and contrastive learning.

The second task is **Text-Image Alignment**. We have two proposed plans for this.

- YOLOv11 [2] and Semantic Approximation: We first use COCO dataset as our train and test dataset. COCO dataset is designed for image understanding with 91 object categories, 328,000 images and 2,500,000 labeled instances. We would pick part of the dataset to train the YOLOv11 models for image extraction. After this, we need to do semantic approximation to match the label with the input text to help the robot arm grasp target object.
- Edge-optimized multi-modal solution: We directly use small models like LLaVA-Phi [3] and TinyLLaVA [4]. We can directly input text and let the model help us identify the target in the image. It can enhance contextual understanding with zero-shot generalization, better suited for out-of-domain (OOD) condition.

2.2.2 Requirements

- Computational Resource: For small models like LLaVA-Phi(2.7B), TinyLLaVA (2.7B) and MobileLLaMa (1.4B), we have RTX 4090 GPU, which is large enough to do inference for the models. Also, we have enough space to save the models.
- Image input: Upon receiving the image, the agent will first distinguish candidate objects from the background. This functionality should be achieved in the base model. After this step, the image should be fragmented into several pieces, with only one object in each piece.
- Embedding and Alignment: We want embedding (for both text and object's image fragment) and the text-image alignment module to be efficient, such that the similarity of correct text-object pair is maximized and the similarity of wrong text-object pair is minimized. The cosine similarity between an object's image fragment embedding $f_I(x_i)$ and a text embedding $f_T(y_j)$ can be formally as:

$$S_{ij} = \frac{f_I(x_i) \cdot f_T(y_j)}{\|f_I(x_i)\| \|f_T(y_j)\|}$$

and this requirement can transform to minimizing the loss function below, where S_{ii} is the similarity between a correct (matching) image-text pair and τ is a temperature parameter that controls the sharpness of the similarity distribution[5].

$$Loss = \frac{1}{2N} \sum_{i=1}^{N} \left[-\log \frac{\exp(S_{ii}/\tau)}{\sum_{j=1}^{N} \exp(S_{ij}/\tau)} - \log \frac{\exp(S_{ii}/\tau)}{\sum_{j=1}^{N} \exp(S_{ji}/\tau)} \right]$$

2.3 Subsystem 2: Robotic Arm

2.3.1 Overview

We plan to use the YahBoom DOFBOT SE Robotic Arm, which is driven by an STM32 controller and uses a virtual machine as the master to generate control decisions. This robotic arm has 6-degree-of-freedom serial bus servo and is controlled by the ROS operating system. By installing a microphone and a USB camera on the outside of the robotic arm, we give the robotic arm visual and auditory perception capabilities. For the end effector, we plan to design a mechanical claw with a maximum opening width of 6 cm and a maximum load of 200 g, so that it can grasp common small objects. Figure 3 shows the specifications of this robotic arm [6].

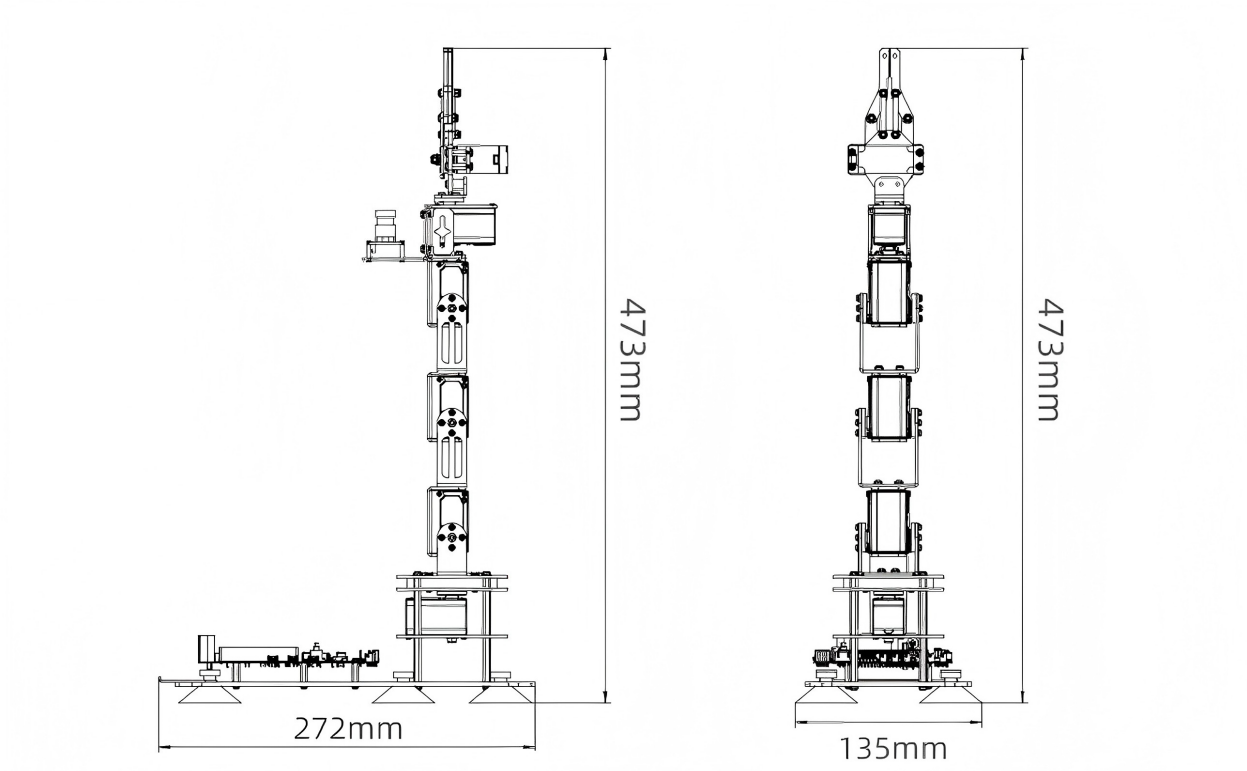

Figure 3: The Specifications Of The Robotic Arm

2.3.2 Requirements

The robotic arm consists of the following components.

- Camera. The Camera subsystem runs continuously, capturing images once every second, and providing real-time visual data to the Image Encoder subsystem for further processing. By operating in a continuous manner, the system can persistently collect and analyze visual information, facilitating rapid detection of changes or anomalies in the environment. Moreover, the steady one-frame per second cap-

ture rate ensures a stable and timely flow of image data, catering to the requirements of subsequent algorithms and monitoring tasks in a broad range of applications.

- ROS environment. Serving as the primary control center, the ROS subsystem oversees hardware operations in accordance with the instructions and outputs produced by the Large Language Model subsystem. By coordinating the execution of robotic tasks, it ensures seamless and precise actions throughout the system. In addition, it forwards the necessary responses to the text-to-speech subsystem, enabling real-time vocal feedback and improving overall operational efficiency.
- The Robotic Arm subsystem features an arm and an end-effector. After receiving the object's coordinates from the vision system, the subsystem employs an inverse kinematics algorithm to determine how each joint should rotate, thereby ensuring accurate movement in three-dimensional space. This algorithm is executed on a PC integrated within a single-chip microcomputer, allowing for real-time, efficient computation. For the grasping mechanism, a clamp is used to hold objects securely. Because the objects to be handled are sufficiently rigid, a force sensor is deemed unnecessary for this design. This simplifies the overall system while still providing reliable, stable grasping performance for tasks requiring object pick-up and placement.
- Microphone. Record audio from users, then convert voice to words for VLM to recognize.
- Path design. This subsystem is designed to receive the target point coordinates of an object from the vision system and utilize an inverse kinematics algorithm to compute the required rotation for each joint. The algorithm will be executed on a PC integrated within a single-chip microcomputer. This approach ensures precise and efficient calculation of joint movements, enabling the robotic system to accurately position and manipulate objects. Additionally, the subsystem will continuously update and adjust the joint angles in real-time to account for any dynamic changes in the object's position, thereby enhancing the overall adaptability and responsiveness of the system.

2.4 Tolerance Analysis

- Data Transfer Analysis Between Computer and Server: Simulations have shown that data transfer delays between the computer and server can be limited to approximately 3-4 seconds. The primary factors affecting this latency include network speed, server processing capabilities, and the complexity of the data. To enhance processing efficiency, we can try to use Python libraries such as flashattention, which significantly accelerate AI model computations. These libraries optimize real-time data analysis and decision-making processes, enabling faster and more efficient handling of information received from user devices. This approach not only reduces latency but also improves the overall responsiveness and accuracy of the system.
- Addressing Overload Errors in Robotic Arm Gripping: An additional challenge

arises when the robotic arm exceeds its maximum load capacity. Although robotic arms are designed with specific weight limits, variations in the weight of handled objects can sometimes result in loads surpassing these limits. When this occurs, the robotic arm may struggle to securely grip the object, leading to errors during the gripping process. To mitigate this issue, a load detection mechanism can be integrated into the control system. This mechanism would detect instances of overload and initiate appropriate responses, such as adjusting the gripping force or applying stabilization techniques, to maintain the stability and accuracy of the gripping operation.

- Optimizing Gripping Strategies for Irregularly Shaped Objects: Irregularities in the shape of objects can pose significant challenges to the gripping accuracy of robotic arms. When handling complex or unevenly shaped items, the robotic arm may struggle to fully conform to the object's surface, resulting in deviations in the gripping position. To overcome this issue, it is essential to optimize the gripping strategy. Techniques such as multi-point gripping or the use of soft grippers can enhance the arm's ability to adapt to various shapes, thereby improving stability and accuracy during the gripping process.

3 Ethics

Given that our target audience primarily includes the elderly and disabled, ethics and safety are crucial aspects of our design. This section is divided into two parts to comprehensively address these concerns.

3.1 Ethics

As ZJUI students, we are committed to upholding ethical standards and ensuring the integrity of our project. Our team will strictly adhere to the IEEE Code of Ethics [7] and the ACM Code of Ethics [8]. We pledge to meet, but are not limited to, the following ethical responsibilities:

- Prioritize public safety, health, and well-being by adhering to ethical design principles and sustainable practices. Additionally, we are obligated to report any potential systemic risks that could lead to harm.
- Strive to benefit society by enhancing individual and collective understanding of both traditional and emerging technologies and their societal implications.
- Maintain honesty and integrity in all professional activities, strictly avoiding unethical conduct such as bribery or other illegal actions.

3.1.1 Safety

To ensure the safety of both team members and others, and to mitigate any potential hazards during the project, our team will strictly comply with the ECE 445 SAFETY GUIDELINES [9]. We will undertake, but are not limited to, the following safety measures:

- No team member is permitted to work alone in the laboratory at any time.
- All team members must complete mandatory safety training before being authorized to work in the laboratory.
- Any handling of battery charging or hazardous battery chemicals must be conducted in strict accordance with established safe usage guidelines.
- The robot must be equipped with an emergency stop mechanism that automatically ceases operation in the event of mechanical failure.

References

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, *Wav2vec 2.0: A framework for self-supervised learning of speech representations*, 2020. arXiv: 2006.11477 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2006.11477.

[2] R. Khanam and M. Hussain, *Yolov11: An overview of the key architectural enhancements*, 2024. arXiv: 2410.17725 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2410.17725.

[3] Y. Zhu, M. Zhu, N. Liu, Z. Ou, X. Mou, and J. Tang, *Llava-phi: Efficient multi-modal assistant with small language model*, 2024. arXiv: 2401.02330 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2401.02330.

[4] B. Zhou, Y. Hu, X. Weng, *et al.*, *Tinyllava: A framework of small-scale large multimodal models*, 2024. arXiv: 2402.14289 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2402.14289.

[5] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.00020.

[6] YahBoom, *Yahboom dofbot se robotic arm specifications*, Accessed: 2025-03-13, 2023. [Online]. Available: https://www.yahboom.com/tbdetails?id=562.

[7] IEEE. ""IEEE Code of Ethics"." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

[8] Association for Computing Machinery, *ACM Code of Ethics and Professional Conduct*, Accessed: 2025-03-13, 2018. [Online]. Available: https://www.acm.org/code-of-ethics.

[9] University of Illinois Urbana-Champaign, *ECE 445 Safety Guidelines*, Accessed: 2025-03-13, 2025. [Online]. Available: https://courses.grainger.illinois.edu/ece445zjui/guidelines/safety.asp.